WHAT IS CLAIMED IS:

1. An XML document editor to enable user to add or delete an element into a working document and to convert said working document into an XML document file; characterized in that said XML document editor automatically generates in relating

5   to two consecutive elements z_i and z_{i+1} of said working document, wherein relation between said elements z_i and z_{i+1} complies with document type definition (DTD) of said document, a list of candidate third element to be alerted to user; wherein said third element z in said list makes relations between elements z_i and z and between elements z and z_{i+1} complying with said DTD, after said element z is inserted

10   between elements z_i and z_{i+1}.

2. The XML document editor according to claim 1, wherein said XML document editor determines whether relation between two consecutive elements comply with said DTD according to the following rule:

suppose G is Glushkov Automaton of said document, z_i is a state in G, $1 \leq i \leq p-1$, p

15     $\in N$, $\Sigma = \{z_1, z_2, ...z_p\}$ is a sequence of states in G where $z_1 = s$, s is start state of G, $z_p = f$, f is final state of G;

if $z_{i+1} \in$ reachable(z_i), wherein reachable(z_i) denote the set of states in G reachable from state z_i,

then the relation between z_i and z_{i+1} is determined compliant with DTD of said

20   document.

3. The XML document editor according to claim 1, wherein said XML generates a cell C to include said candidate third element z according to the following rule and displays said candidates in a list:

suppose $(z_i, z_{i+1}) \notin H$, H denotes the set of edges in G, G is Glushkov Automaton of

25   regular expression E corresponding to an element of said working document; further suppose Σ is a set to include states corresponding to all elements of G, A(E1)

is the set of states in subexpression E1 to E, f-reachable(z_i) denotes the set of states

in G reachable from z_i through forward edges;

if $z_{i+1} \in$ f-reachable(z_i), then let C={ $z \in \Sigma$ | $z \in$ f-reachable(z_i) and $z_{i+1} \in$

f-reachable(z)};

5 if $z_{i+1} \notin$ f-reachable(z_i), then let E1* be the smallest iteration subexpression of E

that covers both z_i and z_{i+1}, C= {$z \in$ A(E1) | $z \in$ f-reachable(z_i) or $z_{i+1} \in$

f-reachable(z)}.

4. The XML document editor according to claim 1, wherein said XML generates a cell

C to include said candidate third element z according to the following rule and

10 displays said candidates in a list:

suppose (z_i, z_{i+1}) $\in$H, H denotes the set of edges in G, G is Glushkov Automaton of

regular expression E corresponding to an element of said working document;

further suppose Σ is a set to include states corresponding to all elements of G, A(E1)

is the set of states in subexpression E1 to E, f-reachable(z_i) denotes the set of states

15 in G reachable from z_i through forward edges;

if (z_i, z_{i+1}) is a forward edge, let C={ $z \in \Sigma$ | $z \in$ f-reachable(z_i) and $z_{i+1} \in$

f-reachable(z) } and:

i) if $z_i \in$ last(E1*) for some iteration subexpression E1* of E and E1 is the

largest one, then let C1= {$z \in$ A(E1) | $z_{i+1} \in$ f-reachable(z)}, C= C $\cup$ C1;

20 ii) If $z_{i+1} \in$ first(E2*) for some iteration subexpression E2* of E and E2 is the

largest one, then let C2= {$z \in$ A(E2) | $z \in$ f-reachable(z_i)} and C= C $\cup$ C2;

if (z_i, z_{i+1}) is a backward edge, then let C= A(E3), wherein E3* is the largest

iteration subexpression of E satisfying $z_i \in$ last(E3) and $z_{i+1} \in$ first(E3).

5. The XML document editor according to claim 1, wherein said XML document

25 editor automatically generates a required element between element pair z_i and z and

element pair z and z_{i+1} after said third element z is inserted between element pair z_i

and z_{i+1}, such that said working document is effective; wherein said requirement comprises articulation points between elements z_i and z (and z and z_{i+1}) in Glushkov Automaton G; i.e., states through which all paths between z_i and z (and z and z_{i+1}) shall pass.

5   6. The XML document editor according to claim 1, wherein said XML document editor automatically generates an element slot allowing user to add elements into said document, if no required element between element pair z_i and z and element pair z and z_{i+1} is found after said third element z is inserted between element pair z_i and z_{i+1} and if $(z_i, z) \notin H$ $((z, z_{i+1}) \notin H)$, wherein H denotes set of edges in G; and

10 wherein said requirement comprises articulation points between elements z_i and z (and z and z_{i+1}) in Glushkov Automaton G; i.e., states through which all paths between z_i and z (and z and z_{i+1}) shall pass.

7. Method for editing an XML document using an XML document editor to enable user to add or delete an element into a working document and to convert said

15 working document into an XML document file; characterized in that said method comprising enabling said XML document editor to automatically generate in relating to two consecutive elements z_i and z_{i+1} of said working document, wherein relation between said elements z_i and z_{i+1} complies with document type definition (DTD) of said document, a list of candidate third element to be alerted to user;

20 wherein said third element z in said list makes relations between elements z_i and z and between elements z and z_{i+1} complying with said DTD, after said element z is inserted between elements z_i and z_{i+1}.

8. The method according to claim 7, wherein whether relation between two consecutive elements complies with said DTD is determined according to the

25 following rule:

suppose G is Glushkov Automaton of said document, z_i is a state in G, $1 \le i \le p-1$, p

$\in N$, $\Sigma = \{z_1, z_2, \ldots z_p\}$ is a sequence of states in G where $z_1=s$, s is start state of G, $z_p=f$, f is final state of G;

if $z_{i+1} \in$ reachable(z_i), wherein reachable(z_i) denote the set of states in G reachable from state z_i,

5   then the relation between z_i and z_{i+1} is determined compliant with DTD of said document.

9.   The method according to claim 7, wherein a cell C to include said candidate third element z is generated according to the following rule and displayed as a list:

suppose (z_i, z_{i+1}) $\notin H$, H denotes the set of edges in G, G is Glushkov Automaton of

10   regular expression E corresponding to an element of said working document;

further suppose Σ is a set to include states corresponding to all elements of G, A(E1) is the set of states in subexpression E1 to E, f-reachable(z_i) denotes the set of states in G reachable from z_i through forward edges;

if $z_{i+1} \in$ f-reachable(z_i), then let C={ $z \in \Sigma$ | $z \in$ f-reachable(z_i) and $z_{i+1} \in$

15   f-reachable(z)};

if $z_{i+1} \notin$ f-reachable(z_i), then let E1* be the smallest iteration subexpression of E that covers both z_i and z_{i+1}, C= {z $\in$ A(E1) | z $\in$ f-reachable(z_i) or $z_{i+1} \in$ f-reachable(z)}.

10. The method according to claim 7, wherein a cell C to include said candidate third

20   element z is generated according to the following rule and displayed as a list:

suppose (z_i, z_{i+1}) $\in H$, H denotes the set of edges in G, G is Glushkov Automaton of regular expression E corresponding to an element of said working document;

further suppose Σ is a set to include states corresponding to all elements of G, A(E1) is the set of states in subexpression E1 to E, f-reachable(z_i) denotes the set of states

25   in G reachable from z_i through forward edges;

if (z_i, z_{i+1}) is a forward edge, let C={ z $\in \Sigma$ | z $\in$ f-reachable(z_i) and $z_{i+1} \in$

f-reachable(z) } and:

iii)   if $z_i \in$ last(E1*) for some iteration subexpression E1* of E and E1 is the largest one, then let C1= {z $\in$ A(E1) | $z_{i+1} \in$ f-reachable(z)}, C= C $\cup$ C1;

iv)   If $z_{i+1} \in$ first(E2*) for some iteration subexpression E2* of E and E2 is the largest one, then let C2= {z $\in$ A(E2) | z $\in$ f-reachable(z_i)} and C= C $\cup$ C2;

if (z_i, z_{i+1}) is a backward edge, then let C= A(E3), wherein E3* is the largest iteration subexpression of E satisfying $z_i \in$ last(E3) and $z_{i+1} \in$ first(E3).

11. The method according to claim 7, further comprising automatically generating a required element between element pair z_i and z and element pair z and z_{i+1} after said third element z is inserted between element pair z_i and z_{i+1}, such that said working document is effective; wherein said requirement comprises articulation points between elements z_i and z (and z and z_{i+1}) in Glushkov Automaton G; i.e., states through which all paths between z_i and z (and z and z_{i+1}) shall pass.

12. The method according to claim 7, further comprising automatically generating an element slot allowing user to add elements into said document, if no required element between element pair z_i and z and element pair z and z_{i+1} is found after said third element z is inserted between element pair z_i and z_{i+1} and if (z_i, z) $\notin$ H ((z, z_{i+1}) $\notin$ H), wherein H denotes set of edges in G; and wherein said requirement comprises articulation points between elements z_i and z (and z and z_{i+1}) in Glushkov Automaton G; i.e., states through which all paths between z_i and z (and z and z_{i+1}) shall pass.

13. An XML document editor, comprising a user interface enabling user to add or delete an element into a working document, whereby said working document is converted into an XML document file; characterized in that said XML document editor automatically generates in relating to two consecutive elements z_i and z_{i+1} of said working document, wherein relation between said elements z_i and z_{i+1}

complies with document type definition (DTD) of said document, a list of candidate

third element to be alerted to usef; wherein said third element z in said list makes

relations between elements z_i and z and between elements z and z_{i+1} complying with

said DTD, after said third element z is inserted between elements z_i and z_{i+1}.

5   14. The XML document editor according to claim 13, wherein said XML document

editor determines whether relation between two consecutive elements comply with

said DTD according to the following rule:

suppose G is Glushkov Automaton of said document, z_i is a state in G, $1 \leq i \leq p-1$, p

$\in$N, $\Sigma = \{z_1, z_2, ...z_p\}$ is a sequence of states in G where $z_1 = s$, s is start state of G,

10    $z_p = f$, f is final state of G;

if $z_{i+1} \in$ reachable(z_i), wherein reachable(z_i) denote the set of states in G reachable

from state z_i,

then the relation between z_i and z_{i+1} is determined compliant with DTD of said

document.

15   15. The XML document editor according to claim 13, wherein said XML generates a

cell C to include said candidate third element z according to the following rule and

displays said candidates in a list:

suppose $(z_i, z_{i+1}) \notin$H, H denotes the set of edges in G, G is Glushkov Automaton of

regular expression E corresponding to an element of said working document;

20   further suppose Σ is a set to include states corresponding to all elements of G, A(E1)

is the set of states in subexpression E1 to E, f-reachable(z_i) denotes the set of states

in G reachable from z_i through forward edges;

if $z_{i+1} \in$ f-reachable(z_i), then let C=$\{$ z $\in \Sigma$ | z $\in$ f-reachable(z_i) and $z_{i+1} \in$

f-reachable(z)$\}$;

25   if $z_{i+1} \notin$ f-reachable(z_i), then let E1* be the smallest iteration subexpression of E

that covers both z_i and z_{i+1}, C= $\{z \in$ A(E1) | z $\in$ f-reachable(z_i) or $z_{i+1} \in$

f-reachable(z)}.

16. The XML document editor according to claim 13, wherein said XML generates a cell C to include said candidate third element z according to the following rule and displays said candidates in a list:

suppose $(z_i, z_{i+1}) \in H$, H denotes the set of edges in G, G is Glushkov Automaton of regular expression E corresponding to an element of said working document; further suppose Σ is a set to include states corresponding to all elements of G, A(E1) is the set of states in subexpression E1 to E, f-reachable(z_i) denotes the set of states in G reachable from z_i through forward edges;

if (z_i, z_{i+1}) is a forward edge, let C={ $z \in \Sigma$ | $z \in$ f-reachable(z_i) and $z_{i+1} \in$ f-reachable(z) } and:

v) if $z_i \in$ last(E1*) for some iteration subexpression E1* of E and E1 is the largest one, then let C1= {z $\in$ A(E1) | $z_{i+1} \in$ f-reachable(z)}, C= C $\cup$ C1;

vi) If $z_{i+1} \in$ first(E2*) for some iteration subexpression E2* of E and E2 is the largest one, then let C2= {z $\in$ A(E2) | z $\in$ f-reachable(z_i)} and C= C $\cup$ C2;

if (z_i, z_{i+1}) is a backward edge, then let C= A(E3), wherein E3* is the largest iteration subexpression of E satisfying $z_i \in$ last(E3) and $z_{i+1} \in$ first(E3).

17. The XML document editor according to claim 13, wherein said XML document editor automatically generates a required element between element pair z_i and z and element pair z and z_{i+1} after said third element z is inserted between element pair z_i and z_{i+1}, such that said working document is effective; wherein said requirement comprises articulation points between elements z_i and z (and z and z_{i+1}) in Glushkov Automaton G; i.e., states through which all paths between z_i and z (and z and z_{i+1}) shall pass.

18. The XML document editor according to claim 13, wherein said XML document editor automatically generates an element slot allowing user to add elements into

said document, if no required element between element pair z_i and z and element

pair z and z_{i+1} is found after said third element z is inserted between element pair z_i

and z_{i+1} and if $(z_i, z) \notin H$ $((z, z_{i+1}) \notin H)$, wherein H denotes set of edges in G; and

wherein said requirement comprises articulation points between elements z_i and z

5   (and z and z_{i+1}) in Glushkov Automaton G; i.e., states through which all paths

between z_i and z (and z and z_{i+1}) shall pass.